

Mobile Content Based Image Retrieval Architectures

Arif Rahman

Information System Department
Universitas Ahmad Dahlan
Yogyakarta, Indonesia
arif.rahman@is.uad.ac.id

Edi Winarko

Comp. Science and Electronics Dept.
Universitas Gadjah Mada
Yogyakarta, Indonesia
ewinarko@ugm.ac.id

Moh. Edi Wibowo

Comp. Science and Electronics Dept.
Universitas Gadjah Mada
Yogyakarta, Indonesia
mediw@ugm.ac.id

Abstract— Mobile device features such as camera and other sensors are evolving rapidly nowadays. Supported by a reliable communications network, it raises new methods in information retrieval. Mobile devices can capture an image with its camera and pass it to the retrieval systems to get the information needed. This system, called Mobile Content-Based Image Retrieval (MCBIR), generally consists of two parts: Offline Database Construction, which create image features database and indexing structure, and Online Image Search, that search images in the database that similar to the user inputs. MCBIR system, based on its computational load and resource needs, can be categorized into three architectural models: client-side, client-server and distributed. These three models were analyzed in three aspects: scalability, latency, and resources. The results show that each architecture has its own characteristics in terms of these aspects and should be considered in the architecture selection phase for MCBIR development.

Keywords—mobile; image; retrieval; content-based; architecture

I. INTRODUCTION

The mobile technology has evolved rapidly in this decade. Mobile devices such as smartphones and tablets are increasing both in terms of a number of users and computing capabilities. The processor's speed has reached the order of GHz, memory capacity has become larger and cheaper and the network connectivity is more varied, i.e. cellular broadband, Wi-Fi, Bluetooth, and Near Field Communication (NFC). The camera with Megapixel resolution has also become a common feature in mobile devices today. With this feature, a mobile device can capture an image and use it as input to search information about surrounding objects. Searching for information based on product photos [1], identifying locations based on landmark photos [2] and reading text from images using OCR (Optical Character Recognition) [3] are some examples of image-based search applications that have been developed in current studies.

Searching an information using images as input often called content-based image retrieval, or mobile visual search [4]. It generally consists of several steps. The image that had taken by a camera then processed to find its similarity with the images in the database. The process is illustrated in Figure 1. [5] [6].

The Offline Database Construction (ODC) is the process of image database construction that will be matched in the searching process. It is called offline because the process is not necessarily connected to a communication network. In this process, important parts of the images in the database are extracted to describe the image features. The popular image feature extraction methods used i.e. SIFT (Scale Invariant

Feature Transform) [7] and SURF (Speeded Up Robust Features) [8]. The features obtained from each image may vary in number, so in the next process, feature coding needs to be done. This process aims to format the features that represent an image in a database, uniformly. Feature Coding can be done with some methods such as Voting-Based Coding or Fisher Coding [9]. Then, codes are stored in an indexing structure such as Inverted file [10].

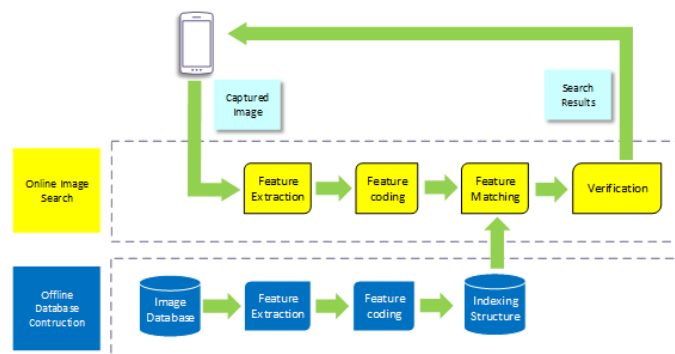


Fig. 1. Mobile content based image retrieval workflows

In Online Image Search (OIS), a user searches through the camera in a mobile device. Captured image extracted its feature and then encoded, similar to the process in ODC. Feature matching process finds the similarity between features obtained from an image query with features in the ODC. Image similarity measurement calculates the distance between those features in certain metrics, such as by Euclidean distance, City Block or Cosine Similarity [11]. Images that have similarity values in certain thresholds are picked as retrieval results. To improve the accuracy, verification of result images is conducted. A method for verification, e.g. RANSAC (Random Sample Consensus [12] is applied to images in search results to obtain the most relevant to the image query.

Mobile devices have the advantage of flexibility and ease of use compared to other computing devices. But mobile devices also have limitations, specifically on computing resources such as CPU performance, memory capacity, and screen size relatives to desktop or laptop computers. This becomes a problem when the volume of data to be processed is large. In MCBIR context, methods have been developed to overcome these limitations such as methods to store and indexing image features efficiently so that it can speed up the process of measuring feature similarities in very large databases [13]. Beside of those factors, MCBIR system architecture can also affect the speed of the retrieval process,

since the architecture determines the computational load between the devices involved. Choosing device resource specifications that fit the computational load makes the MCBIR system run efficiently.

MCBIR system architectures seem not yet specifically discussed in the current literature, so it is necessary to discuss them in this paper. The following section describes several models of MCBIR architectures. Client-side Model in which all retrieval processes are performed on a mobile device, Client-Server model with its variations of process sharing on the pipeline, and to overcome a very large data number, a distributed architecture is developed using multiple servers. Discussion and analysis in scalability, latency and resource dependency of each architecture is presented in the next section. Architecture reviews, discussion, and analysis summarized in the last section of this paper.

II. CLIENT-SIDE MODEL

In this model, all the processes in the image retrieval workflow (ODC and OIS) are performed on mobile devices, as shown in Figure 2. The main process is storing image features efficiently so that can reduce the computational load of the retrieval process. This is due to the limitations of mobile device's computing resources while all processes are to be done on the device. Image features must be compact enough so they can be loaded in relatively small memory on mobile devices and also efficient to compare it in large databases. Also, features should have strong discriminative factors, so that query results have high accuracy [14]. Moreover, feature database should be able to update incrementally, if the user updated or added a new item to the system. For storage and memory efficiency, a compression process is performed, either on the features to be stored or in the indexing file [15], [16].

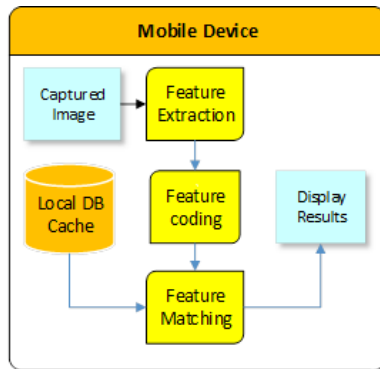


Fig. 2. MCBIR Client Side Model

This model has the advantage of speed retrieval process because it does not depend on the network for data transfer that can cause a delay in the query process and when displaying search results. However, because they are stored on devices with limited capacity compared to PCs or servers, so the size of image data that can be stored is limited. To get compact features to be stored, compressing method is applied to the features or the indexing file.

III. CLIENT-SERVER MODEL

The client-server architecture model is a widely used model in the current image retrieval system. Mobile device becomes a client that captures the image of an object and sends it as the query to the server. Next, computer server analyzes the image, identifies its contents, retrieves relevant information from the database and sends it back to the client. There are three possible configurations in the client-server architecture of MCBIR, i.e. server based, client based and hybrid model [14] [15]

A. Server-based Configuration

In this configuration the computational load on the client is made as light as possible, there is no feature extraction and other processes except send the query image and display the query results that the server sends as a response. So most of the process is performed on the server. In the searching process, client sends an original or scaled down image to server as shown in Figure 3

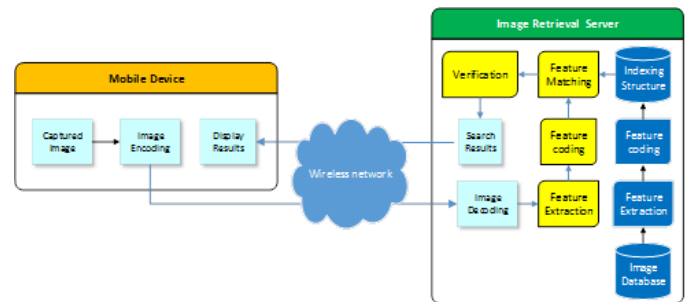


Fig. 3. Server-based Configuration

Image encoding aims to reduce image data to be sent to the server and can be done by minimizing the image size or image compression (JPEG / PNG). ODC and OIS processes are performed on the server with input from the encoded image in the client. The advantage of this model is the hardware resource specification required by the client is relatively low because the computational load for the retrieval process in the client is not complicated. However, the server load is relatively high as the overall retrieval process is performed on the server and also the load will increase as the client connects to the server.

B. Client-based Configuration

The client-based configuration intends to reduce the computing load of the server. Unlike the server based configuration, in this configuration, the query image on the mobile device will be extracted and encoded before being sent to the server. Figure 3 describes the configuration.

The extraction and encoding of image features are performed before being sent to the server. It aims to minimize the data sent from mobile devices and also reduces server load during OIS process. The processes performed on the server are ODC and some of OIS i.e. feature matching and verification, based on the results of encoding features that are sent from mobile devices. The server-side computing load on this model

is lower than client-based, but the client's resource requirements are higher, due to increasing computational loads.

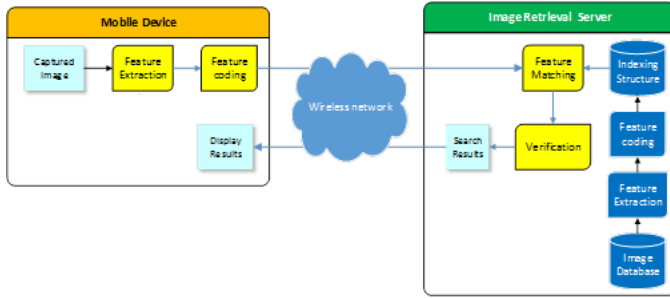


Fig. 4. Client-based Configuration

C. Hybrid Configuration

This configuration might reduce the client's communication time to the server by reducing the query request to the server. This can be done using a local database or cache. The purpose of local database is to store image features that have been provided by the system or have been accessed by the client, so that feature matching process can be performed in the client, as shown in Figure 5

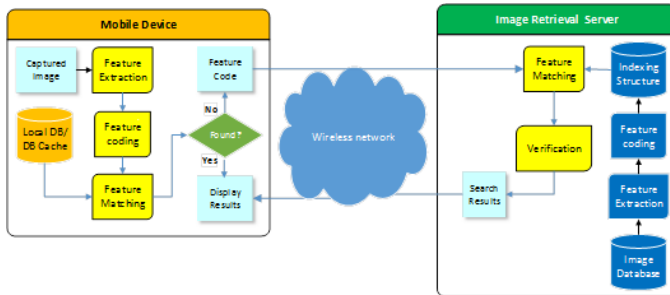


Fig. 5. Hybrid Configuration

When the system searches in the local database and gets results that are relevant to image query, then the searching is stopped and then the results are displayed. Otherwise, if there are no images relevant to image query in local database, then client will send the query to the server which has a larger image database, to perform retrieval process and send the results to the client [16]

IV. DISTRIBUTED MODEL

In the client-server model, problems will arise when there is a need for large amounts of image data storage, causing server-side processing sometimes unable to be handled by a single server. This can be overcome by providing more than one server that can work together to manage resources or distributed systems. CBIR systems using this approach include [17] and [18]

There are two different ways of partitioning in a distributed CBIR system: partitioning by index or document. On index-based partitioning, the features of the entire images in database are extracted, indexed and stored on one machine first, then the index is split into parts to be distributed to multiple servers

But those methods have drawbacks, in the initial step of index construction is performed on a single machine, so if the data are very large, then resource limitations problem will arise. Another problem is when a new image document is added, it must be re-indexed by adding a newly inserted image index and redistributing it to each server which is time costly.

In document based partitioning, documents and indexes are distributed to each server or using local indexes on each server. Consequently, adding new image document does not affect overall index of the distributed system. However, those partitioning methods require a global index that can be recognized across servers. If there is no such global index provided, a strategy is needed to rank and merge query results from each server. Generally distributed models in MCBIR are shown in Figure 6

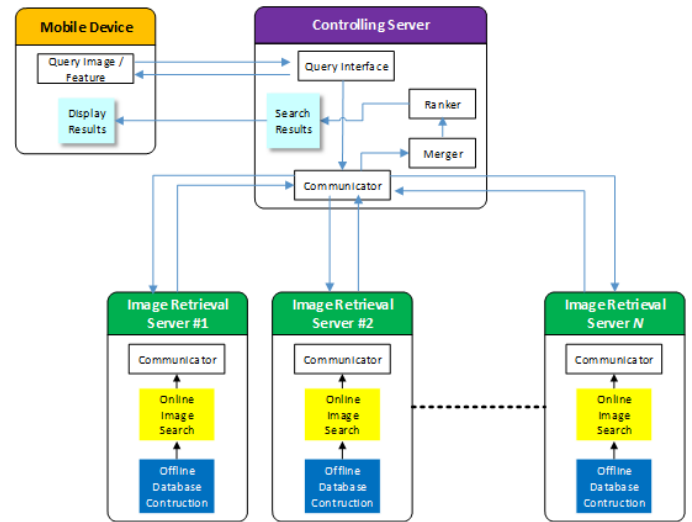


Fig. 6. Distributed Model

Controlling servers receive and process queries sent from mobile devices, then distribute them to image search servers. Each server in the distributed model works like a server in the client-server model. The results from each server merged and ranked in the controlling server. Communication between servers via communicator interface is using standardized communication and routing protocols.

Distributed model has advantage in data search scalability for huge amount of data, but has a disadvantage that is highly dependent on the network and requires additional process and time for communication between servers

V. DISCUSSION

MCBIR architecture models described in the previous sections can be analyzed and categorized based on the scalability of the data that can be processed, latency or response time, and resources used as shown in Table 1. The categorization is based on the assumption of maximum capability of each architecture relative to other architectural models.

TABLE I. ANALYSIS OF MCBIR ARCHITECTURE MODELS

Model	Scalability	Latency	Resources		
			Client	Server	Network
Client-side	small	low	high	-	-
Client-server	medium	medium	low-medium	medium-high	high
Distributed	large	high	low-medium	high	high

Scalability is categorized based on the capability of hardware and technology to handle database processing. Distributed models can handle the largest database compared to other models since it has multiple servers for storing documents and processing queries, contrasts to the client-side model which has limited resources on a single device. However, in terms of latency, the client model has the fastest response since it does not need a communication network to transfer the query image and retrieval results to another device. In the client-server model, resources that needed by its components have more variety than other models since it has different configurations that distribute computational load between client and server.

MCBIR system with the client model becomes more reliable today since it has sophisticated features, enabling better recognition capabilities as it can store larger databases. Similarly, with the evolution of server technology such as virtualization, allows single server machine to have more scalability as multiple servers have. Moreover, the recent network technology that can transfer data faster, enabling distributed models to search on a larger database efficiently.

Table 1 can be considered in MCBIR architecture model selection in terms of scalability, response time and resource dependencies.

VI. CONCLUSION

Each architecture has its characteristics as discussed in the previous sections. Client-side model is suitable for fast response system with small database while client-server has various configurations for flexibility in load balancing between client and server. Scalability problem on very large image database system is the concern of a distributed system to overcome it. These characteristics can be considered in architecture model selection phase on MCBIR system development

The evolution of mobile devices, servers, and network technologies can also affect MCBIR architecture models. In other words, MCBIR architecture will continue to evolve to adapt the evolution of its component technology.

REFERENCES

- [1] X. Shen, Z. Lin, J. Brandt, and Y. Wu, "Mobile product image search by automatic query object extraction," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 7575 LNCS, no. PART 4, pp. 114–127, 2012.
- [2] D. Li and M. C. Chuah, "EMOVIS: An Efficient Mobile Visual Search System for Landmark Recognition," in 2013 IEEE 9th International Conference on Mobile Ad-hoc and Sensor Networks, 2013, pp. 53–60.
- [3] M. Zhang, A. Joshi, R. Kadmawala, K. Dantu, S. Poduri, and G. S. Sukhatme, "OCRdroid: A framework to digitize text using mobile phones," Lect. Notes Inst. Comput. Sci. Soc. Telecommun. Eng., vol. 35 LNICST, pp. 273–292, 2010.
- [4] B. Girod, V. Chandrasekhar, D. M. Chen, N. M. Cheung, R. Grzeszczuk, Y. Reznik, G. Takacs, S. S. Tsai, and R. Vedantham, "Mobile visual search," IEEE Signal Process. Mag., vol. 28, no. 4, pp. 61–76, 2011.
- [5] G. Hua and X. S. Hua, "Mobile cloud visual media computing: From interaction to service," Mob. Cloud Vis. Media Comput. From Interact. to Serv., pp. 1–353, 2015.
- [6] N. Cheung, R. Grzeszczuk, Y. Reznik, and S. Member, "Mobile Visual Search," pp. 1–11, 2013.
- [7] D. G. Lowe, "Distinctive image features from scale invariant keypoints," Int'l J. Comput. Vis., vol. 60, pp. 91–110, 2004.
- [8] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 3951 LNCS, pp. 404–417, 2006.
- [9] Y. Huang, Z. Wu, L. Wang, and S. Member, "Feature Coding in Image Classification," vol. 36, no. 3, pp. 493–506, 2014.
- [10] J. Sivic and A. Zisserman, "Video Google: a text retrieval approach to object matching in videos," Proc. Int. Conf. Comput. Vis., vol. 2, pp. 1470–1477, 2003.
- [11] J. Collins and K. Okada, "A Comparative Study of Similarity Measures for Content-Based Medical Image Retrieval."
- [12] M. a. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," Commun. ACM, vol. 24, no. 6, pp. 381–395, 1981.
- [13] G. Hua and X. S. Hua, "Mobile cloud visual media computing: From interaction to service," 2015.
- [14] A. Vetro and B. Girod, "Industry and Standards Mobile Visual Search : Architectures , Technologies , and the Emerging MPEG Standard," pp. 86–94, 2011.
- [15] D. L. Epfl and C.- Lausanne, "ON OPTIMAL SOLUTIONS FOR MOBILE IMAGE RETRIEVAL APPLICATIONS Multimedia Signal Processing Group – MMSPG Institute of Electrical Engineering – IEL Ecole Polytechnique F '," pp. 9–10.
- [16] D. M. Chen and B. Girod, "With Compact Global Signatures," vol. 17, no. 7, pp. 1019–1030, 2015.
- [17] W. Wang, Y. Wu, and A. Zhang, "SemView: a semantic-sensitive distributed image retrieval system," dg.o '03 Proc. 2003 Annu. Natl. Conf. Digit. Gov. Res., vol. 1, pp. 1–4, 2003.
- [18] Y. Z. Y. Zheng, X. X. X. Xie, and W.-Y. M. W.-Y. Ma, "Distributed Architecture for Large Scale Image-Based Search," Multimed. Expo, 2007 IEEE Int. Conf., no. 49, pp. 579–582, 2007.